PORTAL
US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

orb to orb message fragment offset variable zero performance

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used
orb to orb message fragment offset variable zero performance

Found **41,978** of **145,831**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale □ ▨ ▨ ▨ ■

1 Automatic data and computation decomposition on distributed memory parallel computers ■

Peizong Lee, Zvi Meir Kedem

January 2002 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 24 Issue 1

Full text available: pdf(1.15 MB) Additional Information: full citation, abstract, references, index terms

To exploit parallelism on shared memory parallel computers (SMPCs), it is natural to focus on decomposing the computation (mainly by distributing the iterations of the nested Do-Loops). In contrast, on distributed memory parallel computers (DMPCs), the decomposition of computation and the distribution of data must both be handled---in order to balance the computation load and to minimize the migration of data. We propose and validate experimentally a method for handling computations and data syn ...

Keywords: Computation decomposition, data alignment, data distribution, distributed-memory computers, dominant data array, iteration space mapping vector, parallelizing compilers, spatial dependence vector, temporal dependence vector, tiling techniques

2 Flick: a flexible, optimizing IDL compiler ■

Eric Eide, Kevin Frei, Bryan Ford, Jay Lepreau, Gary Lindstrom

May 1997 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1997 conference on Programming language design and implementation**, Volume 32 Issue 5

Full text available: pdf(1.75 MB) Additional Information: full citation, abstract, references, citings, index terms

An interface definition language (IDL) is a nontraditional language for describing interfaces between software components. IDL compilers generate "stubs" that provide separate communicating processes with the abstraction of local object invocation or procedure call. High-quality stub generation is essential for applications to benefit from component-based designs, whether the components reside on a single computer or on multiple networked hosts. Typical IDL compilers, ...

3 Fast detection of communication patterns in distributed executions ▨

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

4 The design and performance of a pluggable protocols framework for real-time distributed object computing middleware
Carlos O'Ryan, Fred Kuhns, Douglas C. Schmidt, Ossama Othman, Jeff Parsons
April 2000 **IFIP/ACM International Conference on Distributed systems platforms**

Full text available: pdf(231.64 KB) Additional Information: full citation, abstract, references, citings

To be an effective platform for performance-sensitive real-time and embedded applications, off-the-shelf CORBA middleware must preserve the communication-layer quality of service (QoS) properties of applications end-to-end. However, the standard CORBA GIOP/HOP interoperability protocols are not well suited for applications that cannot tolerate the message footprint size, latency, and jitter associated with general-purpose messaging and transport protocols. It is essential, therefore, to de ...

5 OBJEKT—a persistent object store with an integrated garbage collector
D M Harland, B Beloff
April 1987 **ACM SIGPLAN Notices**, Volume 22 Issue 4

Full text available: pdf(1.06 MB) Additional Information: full citation, abstract, index terms

This paper describes OBJEKT, a single-level persistent storage system designed for the REKURSIV architecture.It will be shown that OBJEKT can be microcoded to implement "objects" efficiently, and that data integrity can be guaranteed by provision of an object oriented instruction set. Particular attention will be paid to its facilities for type and range checking, to its object by- object paging strategy and to ways of enhancing parallelism during garbage collection.

6 Measuring the performance of communication middleware on high-speed networks
Aniruddha Gokhale, Douglas C. Schmidt
August 1996 **ACM SIGCOMM Computer Communication Review , Conference proceedings on Applications, technologies, architectures, and protocols for computer communications**, Volume 26 Issue 4

Full text available: pdf(270.13 KB) Additional Information: full citation, abstract, references, citings, index terms

Conventional implementations of communication middleware (such as CORBA and traditional RPC toolkits) incur considerable over-head when used for performance-sensitive applications over high-speed networks. As gigabit networks become pervasive, inefficient middleware will force programmers to use lower-level mechanisms to achieve the necessary transfer rates. This is a serious problem for mission/life-critical applications (such as satellite surveillance and medical imaging).This paper compares t ...

7 The design and performance of a scable ORB architecture for COBRA asynchronous messaging
Alexander B. Arulanthu, Carlos O'Ryan, Douglas C. Schmidt, Michael Kircher, Jeff Parsons
April 2000 **IFIP/ACM International Conference on Distributed systems platforms**

Full text available: pdf(174.72 KB) Additional Information: full citation, abstract, references, citings

Historically, method-oriented middleware, such as Sun RPC, DCE, Java RMI, COM, and

CORBA, has provided synchronous method invocation (SMI) models to applications. Although SMI works well for conventional client/server applications, it is not well-suited for high-performance or real-time applications due to its lack of scalability. To address this problem, the OMG has recently standardized an asynchronous method invocation (AMI) model for CORBA. AMI provides CORBA with many of the capabilit ...

8 Middleware performance analysis: Application level perfomance optimizations for CORBA-based systems
Weili Tao, Shikharesh Majumdar
July 2002 **Proceedings of the third international workshop on Software and performance**
Full text available: pdf(89.30 KB) Additional Information: full citation, abstract, references, citings

Middleware provides inter-operability in a heterogeneous distributed object computing environment. Common Object Request Broker (CORBA) is a standard for middleware proposed by OMG. Although inter-operability is achieved middleware often introduces overheads that impair system performance. This research is concerned with performance enhancement of CORBA-based systems by deploying appropriate techniques at the application level. The paper demonstrates that decisions made by the application softwa ...

Keywords: CORBA performance, design guidelines, middleware performance, performance optimization

9 Profiling and reducing processing overheads in TCP/IP
Jonathan Kay, Joseph Pasquale
December 1996 **IEEE/ACM Transactions on Networking (TON)**, Volume 4 Issue 6
Full text available: pdf(1.21 MB) Additional Information: full citation, references, citings, index terms

10 The Proteus multiprotocol message library
Kenneth Chiu, Madhusudhan Govindaraju, Dennis Gannon
November 2002 **Proceedings of the 2002 ACM/IEEE conference on Supercomputing**
Full text available: pdf(128.51 KB) Additional Information: full citation, abstract, references, citings, index terms

Grid systems span manifold organizations and application domains. Because this diverse environment inevitably engenders multiple protocols, interoperability mechanisms are crucial to seamless, pervasive access. This paper presents the design, rationale, and implementation of the Proteus multiprotocol library for integrating multiple message protocols, such as SOAP and JMS, within one system. Proteus decouples application code from protocol code at run-time, allowing clients to incorporate separa ...

Keywords: SOAP, component, grid, middleware, multiprotocol

11 PCCTS reference manual: version 1.00
T. J. Parr, H. G. Dietz, W. E. Cohen
February 1992 **ACM SIGPLAN Notices**, Volume 27 Issue 2
Full text available: pdf(3.77 MB) Additional Information: full citation, citings, index terms

12 Aspects in the middle: Measuring the dynamic behaviour of AspectJ programs
Bruno Dufour, Christopher Goard, Laurie Hendren, Oege de Moor, Ganesh Sittampalam, Clark

Verbrugge ·
October 2004 **Proceedings of the 19th annual ACM SIGPLAN Conference on Object-oriented programming, systems, languages, and applications**
Full text available: pdf(226.86 KB) Additional Information: full citation, abstract, references, index terms

This paper proposes and implements a rigorous method for studying the dynamic behaviour of AspectJ programs. As part of this methodology several new metrics specific to AspectJ programs are proposed and tools for collecting the relevant metrics are presented. The major tools consist of: (1) a modified version of the AspectJ compiler that tags bytecode instructions with an indication of the cause of their generation, such as a particular feature of AspectJ; and (2) a modified version of the *J ...

Keywords: AspectJ, aspect-oriented programming, dynamic metrics, java, optimization, performance, program analysis

13 Taming the IXP network processor
Lal George, Matthias Blume
May 2003 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation**, Volume 38 Issue 5
Full text available: pdf(159.27 KB) Additional Information: full citation, abstract, references, citings, index terms, review

We compile Nova, a new language designed for writing network processing applications, using a back end based on integer-linear programming (ILP) for register allocation, optimal bank assignment, and spills. The compiler's optimizer employs CPS as its intermediate representation; some of the invariants that this IR guarantees are essential for the formulation of a practical ILP model.Appel and George used a similar ILP-based technique for the IA32 to decide which variables reside in registers but ...

Keywords: Intel IXA, bank assignment, code generation, integer linear programming, network processors, programming languages, register allocation

14 Monitoring, security, and dynamic configuration with the dynamicTAO reflective ORB
Fabio Kon, Manuel Román, Ping Liu, Jina Mao, Tomonori Yamane, Claudio Magalhã, Roy H. Campbell
April 2000 **IFIP/ACM International Conference on Distributed systems platforms**
Full text available: pdf(432.36 KB) Additional Information: full citation, abstract, references, citings

Conventional middleware systems fail to address important issues related to dynamism. Modern computer systems have to deal not only with heterogeneity in the underlying hardware and software platforms but also with highly dynamic environments. Mobile and distributed applications are greatly affected by dynamic changes of the environment characteristic such as security constraints and resource availability. Existing middleware is not prepared to react to these changes. In many cases, applicati ...

15 Technical correspondence: Vector pascal reference manual
Paul Cockshott
June 2002 **ACM SIGPLAN Notices**, Volume 37 Issue 6
Full text available: pdf(1.81 MB) Additional Information: full citation, references

16 System design: Exploiting prescriptive aspects: a design time capability
John A. Stankovic, Prashant Nagaraddi, Zhendong Yu, Zhimin He, Brian Ellis
September 2004 **Proceedings of the fourth ACM international conference on Embedded**

software

Full text available: pdf(353.75 KB) Additional Information: full citation, abstract, references, index terms

Aspect oriented programming (AOP), when used well, has many advantages. Aspects are however, programming-time constructs, i.e., they relate to source code. Previously, we developed a tool called VEST that extended aspects to design time for embedded systems. Two types of design time aspects were identified which we labeled aspect checks and prescriptive aspects. In the original VEST tool several keys aspect checks and a simple form of prescriptive aspects were implemented. Prescriptive aspects a ...

Keywords: aspects, component-based design, prescriptive aspects

17 Implications of hierarchical N-body methods for multiprocessor architectures
Jaswinder Pal Singh, John L. Hennessy, Anoop Gupta
May 1995 **ACM Transactions on Computer Systems (TOCS)**, Volume 13 Issue 2

Full text available: pdf(4.66 MB) Additional Information: full citation, abstract, references, citings, index terms, review

To design effective large-scale multiprocessors, designers need to understand the characteristics of the applications that will use the machines. Application characteristics of particular interest include the amount of communication relative to computation, the structure of the communication, and the local cache and memory requirements, as well as how these characteristics scale with larger problems and machines. One important class of applications is based on hierarchical N-body methods, w ...

Keywords: N-body methods, communication abstractions, locality, message passing, parallel applications, parallel computer architecture, scaling, shared address space, shared memory

18 Middleware performance analysis: Performance monitoring of java applications
M. Harkema, D. Quartel, B. M. M. Gijsen, R. D. van der Mei
July 2002 **Proceedings of the third international workshop on Software and performance**

Full text available: pdf(219.69 KB) Additional Information: full citation, abstract, references, citings, index terms

Over the past few years, Java has evolved into a mature platform for developing enterprise applications. A critical factor for the commercial success of these applications is end-to-end performance, e.g., in terms of response times, throughput and availability. This raises the need for the development, validation and analysis of performance models to predict performance metrics of interest. To develop and validate performance models, insight in the execution behavior of the application is essent ...

Keywords: performance measurement and monitoring of java applications

19 Fragmentation considered harmful
Christopher A. Kent, Jeffrey C. Mogul
January 1995 **ACM SIGCOMM Computer Communication Review**, Volume 25 Issue 1

Full text available: pdf(1.25 MB) Additional Information: full citation, abstract, index terms

Internetworks can be built from many different kinds of networks, with varying limits on maximum packet size. Throughput is usually maximized when the largest possible packet is sent; unfortunately, some routes can carry only very small packets. The IP protocol allows a gateway to *fragment* a packet if it is too large to be transmitted. Fragmentation is at best a necessary evil; it can lead to poor performance or complete communication failure. There are a variety of ways to reduce the lik ...

20 A collaborative framework for distributed microscopy
B. Parvin, J. Taylor, G. Cong
November 1998 **Proceedings of the 1998 ACM/IEEE conference on Supercomputing (CDROM)**
Full text available: pdf(613.03 KB) Additional Information: full citation, abstract, references, citings

This paper outlines the motivation, requirements, and architecture of a collaborative framework for distributed virtual microscopy. In this context, the requirements are specified in terms of (1) functionality, (2) scalability, (3) interactivity, and (4) safety and security. Functionality refers to what and how an instrument does something. Scalability refers to the number of instruments, vendor-specific desktop workstations, analysis programs, and collaborators that can be accessed. Interactivi ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

PORTAL

US Patent & Trademark Office

Subscribe (Full Service) Register (Limited Service, Free) Login

Search: ● The ACM Digital Library ○ The Guide

orb to orb message fragment offset variable zero performance

THE ACM DIGITAL LIBRARY

Feedback Report a problem Satisfaction survey

Terms used
orb to orb message fragment offset variable zero performance

Found **41,978** of **145,831**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

□ Open results in a new window

Try an Advanced Search
Try this search in The ACM Guide

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

1 Automatic data and computation decomposition on distributed memory parallel computers

Peizong Lee, Zvi Meir Kedem

January 2002 **ACM Transactions on Programming Languages and Systems (TOPLAS)**, Volume 24 Issue 1

Full text available: pdf(1.15 MB) Additional Information: full citation, abstract, references, index terms

To exploit parallelism on shared memory parallel computers (SMPCs), it is natural to focus on decomposing the computation (mainly by distributing the iterations of the nested Do-Loops). In contrast, on distributed memory parallel computers (DMPCs), the decomposition of computation and the distribution of data must both be handled---in order to balance the computation load and to minimize the migration of data. We propose and validate experimentally a method for handling computations and data syn ...

Keywords: Computation decomposition, data alignment, data distribution, distributed-memory computers, dominant data array, iteration space mapping vector, parallelizing compilers, spatial dependence vector, temporal dependence vector, tiling techniques

2 Flick: a flexible, optimizing IDL compiler

Eric Eide, Kevin Frei, Bryan Ford, Jay Lepreau, Gary Lindstrom

May 1997 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1997 conference on Programming language design and implementation**, Volume 32 Issue 5

Full text available: pdf(1.75 MB) Additional Information: full citation, abstract, references, citings, index terms

An interface definition language (IDL) is a nontraditional language for describing interfaces between software components. IDL compilers generate "stubs" that provide separate communicating processes with the abstraction of local object invocation or procedure call. High-quality stub generation is essential for applications to benefit from component-based designs, whether the components reside on a single computer or on multiple networked hosts. Typical IDL compilers, ...

3 Fast detection of communication patterns in distributed executions

Thomas Kunz, Michiel F. H. Seuren

November 1997 **Proceedings of the 1997 conference of the Centre for Advanced Studies on Collaborative research**

Full text available: pdf(4.21 MB) Additional Information: full citation, abstract, references, index terms

Understanding distributed applications is a tedious and difficult task. Visualizations based on process-time diagrams are often used to obtain a better understanding of the execution of the application. The visualization tool we use is Poet, an event tracer developed at the University of Waterloo. However, these diagrams are often very complex and do not provide the user with the desired overview of the application. In our experience, such tools display repeated occurrences of non-trivial commun ...

4 The design and performance of a pluggable protocols framework for real-time distributed object computing middleware
Carlos O'Ryan, Fred Kuhns, Douglas C. Schmidt, Ossama Othman, Jeff Parsons
April 2000 **IFIP/ACM International Conference on Distributed systems platforms**

Full text available: pdf(231.64 KB) Additional Information: full citation, abstract, references, citings

To be an effective platform for performance-sensitive real-time and embedded applications, off-the-shelf CORBA middleware must preserve the communication-layer quality of service (QoS) properties of applications end-to-end. However, the standard CORBA GIOP/HOP interoperability protocols are not well suited for applications that cannot tolerate the message footprint size, latency, and jitter associated with general-purpose messaging and transport protocols. It is essential, therefore, to de ...

5 OBJEKT—a persistent object store with an integrated garbage collector
D M Harland, B Beloff
April 1987 **ACM SIGPLAN Notices**, Volume 22 Issue 4

Full text available: pdf(1.06 MB) Additional Information: full citation, abstract, index terms

This paper describes OBJEKT, a single-level persistent storage system designed for the REKURSIV architecture.It will be shown that OBJEKT can be microcoded to implement "objects" efficiently, and that data integrity can be guaranteed by provision of an object oriented instruction set. Particular attention will be paid to its facilities for type and range checking, to its object by- object paging strategy and to ways of enhancing parallelism during garbage collection.

6 Measuring the performance of communication middleware on high-speed networks
Aniruddha Gokhale, Douglas C. Schmidt
August 1996 **ACM SIGCOMM Computer Communication Review , Conference proceedings on Applications, technologies, architectures, and protocols for computer communications**, Volume 26 Issue 4

Full text available: pdf(270.13 KB) Additional Information: full citation, abstract, references, citings, index terms

Conventional implementations of communication middleware (such as CORBA and traditional RPC toolkits) incur considerable over-head when used for performance-sensitive applications over high-speed networks. As gigabit networks become pervasive, inefficient middleware will force programmers to use lower-level mechanisms to achieve the necessary transfer rates. This is a serious problem for mission/life-critical applications (such as satellite surveillance and medical imaging).This paper compares t ...

7 The design and performance of a scable ORB architecture for COBRA asynchronous messaging
Alexander B. Arulanthu, Carlos O'Ryan, Douglas C. Schmidt, Michael Kircher, Jeff Parsons
April 2000 **IFIP/ACM International Conference on Distributed systems platforms**

Full text available: pdf(174.72 KB) Additional Information: full citation, abstract, references, citings

Historically, method-oriented middleware, such as Sun RPC, DCE, Java RMI, COM, and

CORBA, has provided synchronous method invocation (SMI) models to applications. Although SMI works well for conventional client/server applications, it is not well-suited for high-performance or real-time applications due to its lack of scalability. To address this problem, the OMG has recently standardized an asynchronous method invocation (AMI) model for CORBA. AMI provides CORBA with many of the capabilit ...

8 Middleware performance analysis: Application level performance optimizations for CORBA-based systems
Weili Tao, Shikharesh Majumdar
July 2002 **Proceedings of the third international workshop on Software and performance**
Full text available: pdf(89.30 KB) Additional Information: full citation, abstract, references, citings

Middleware provides inter-operability in a heterogeneous distributed object computing environment. Common Object Request Broker (CORBA) is a standard for middleware proposed by OMG. Although inter-operability is achieved middleware often introduces overheads that impair system performance. This research is concerned with performance enhancement of CORBA-based systems by deploying appropriate techniques at the application level. The paper demonstrates that decisions made by the application softwa ...

Keywords: CORBA performance, design guidelines, middleware performance, performance optimization

9 Profiling and reducing processing overheads in TCP/IP
Jonathan Kay, Joseph Pasquale
December 1996 **IEEE/ACM Transactions on Networking (TON)**, Volume 4 Issue 6
Full text available: pdf(1.21 MB) Additional Information: full citation, references, citings, index terms

10 The Proteus multiprotocol message library
Kenneth Chiu, Madhusudhan Govindaraju, Dennis Gannon
November 2002 **Proceedings of the 2002 ACM/IEEE conference on Supercomputing**
Full text available: pdf(128.51 KB) Additional Information: full citation, abstract, references, citings, index terms

Grid systems span manifold organizations and application domains. Because this diverse environment inevitably engenders multiple protocols, interoperability mechanisms are crucial to seamless, pervasive access. This paper presents the design, rationale, and implementation of the Proteus multiprotocol library for integrating multiple message protocols, such as SOAP and JMS, within one system. Proteus decouples application code from protocol code at run-time, allowing clients to incorporate separa ...

Keywords: SOAP, component, grid, middleware, multiprotocol

11 PCCTS reference manual: version 1.00
T. J. Parr, H. G. Dietz, W. E. Cohen
February 1992 **ACM SIGPLAN Notices**, Volume 27 Issue 2
Full text available: pdf(3.77 MB) Additional Information: full citation, citings, index terms

12 Aspects in the middle: Measuring the dynamic behaviour of AspectJ programs
Bruno Dufour, Christopher Goard, Laurie Hendren, Oege de Moor, Ganesh Sittampalam, Clark

Verbrugge

October 2004 **Proceedings of the 19th annual ACM SIGPLAN Conference on Object-oriented programming, systems, languages, and applications**

Full text available: pdf(226.86 KB) Additional Information: full citation, abstract, references, index terms

This paper proposes and implements a rigorous method for studying the dynamic behaviour of AspectJ programs. As part of this methodology several new metrics specific to AspectJ programs are proposed and tools for collecting the relevant metrics are presented. The major tools consist of: (1) a modified version of the AspectJ compiler that tags bytecode instructions with an indication of the cause of their generation, such as a particular feature of AspectJ; and (2) a modified version of the *J ...

Keywords: AspectJ, aspect-oriented programming, dynamic metrics, java, optimization, performance, program analysis

13 Taming the IXP network processor

Lal George, Matthias Blume

May 2003 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation**, Volume 38 Issue 5

Full text available: pdf(159.27 KB) Additional Information: full citation, abstract, references, citings, index terms, review

We compile Nova, a new language designed for writing network processing applications, using a back end based on integer-linear programming (ILP) for register allocation, optimal bank assignment, and spills. The compiler's optimizer employs CPS as its intermediate representation; some of the invariants that this IR guarantees are essential for the formulation of a practical ILP model.Appel and George used a similar ILP-based technique for the IA32 to decide which variables reside in registers but ...

Keywords: Intel IXA, bank assignment, code generation, integer linear programming, network processors, programming languages, register allocation

14 Monitoring, security, and dynamic configuration with the dynamicTAO reflective ORB

Fabio Kon, Manuel Román, Ping Liu, Jina Mao, Tomonori Yamane, Claudio Magalhã, Roy H. Campbell

April 2000 **IFIP/ACM International Conference on Distributed systems platforms**

Full text available: pdf(432.36 KB) Additional Information: full citation, abstract, references, citings

Conventional middleware systems fail to address important issues related to dynamism. Modern computer systems have to deal not only with heterogeneity in the underlying hardware and software platforms but also with highly dynamic environments. Mobile and distributed applications are greatly affected by dynamic changes of the environment characteristic such as security constraints and resource availability. Existing middleware is not prepared to react to these changes. In many cases, applicati ...

15 Technical correspondence: Vector pascal reference manual

Paul Cockshott

June 2002 **ACM SIGPLAN Notices**, Volume 37 Issue 6

Full text available: pdf(1.81 MB) Additional Information: full citation, references

16 System design: Exploiting prescriptive aspects: a design time capability

John A. Stankovic, Prashant Nagaraddi, Zhendong Yu, Zhimin He, Brian Ellis

September 2004 **Proceedings of the fourth ACM international conference on Embedded**

software

Full text available: pdf(353.75 KB) Additional Information: full citation, abstract, references, index terms

Aspect oriented programming (AOP), when used well, has many advantages. Aspects are however, programming-time constructs, i.e., they relate to source code. Previously, we developed a tool called VEST that extended aspects to design time for embedded systems. Two types of design time aspects were identified which we labeled aspect checks and prescriptive aspects. In the original VEST tool several keys aspect checks and a simple form of prescriptive aspects were implemented. Prescriptive aspects a ...

Keywords: aspects, component-based design, prescriptive aspects

17 Implications of hierarchical N-body methods for multiprocessor architectures
Jaswinder Pal Singh, John L. Hennessy, Anoop Gupta
May 1995 **ACM Transactions on Computer Systems (TOCS)**, Volume 13 Issue 2

Full text available: pdf(4.66 MB) Additional Information: full citation, abstract, references, citings, index terms, review

To design effective large-scale multiprocessors, designers need to understand the characteristics of the applications that will use the machines. Application characteristics of particular interest include the amount of communication relative to computation, the structure of the communication, and the local cache and memory requirements, as well as how these characteristics scale with larger problems and machines. One important class of applications is based on hierarchical N-body methods, w ...

Keywords: N-body methods, communication abstractions, locality, message passing, parallel applications, parallel computer architecture, scaling, shared address space, shared memory

18 Middleware performance analysis: Performance monitoring of java applications
M. Harkema, D. Quartel, B. M. M. Gijsen, R. D. van der Mei
July 2002 **Proceedings of the third international workshop on Software and performance**

Full text available: pdf(219.69 KB) Additional Information: full citation, abstract, references, citings, index terms

Over the past few years, Java has evolved into a mature platform for developing enterprise applications. A critical factor for the commercial success of these applications is end-to-end performance, e.g., in terms of response times, throughput and availability. This raises the need for the development, validation and analysis of performance models to predict performance metrics of interest. To develop and validate performance models, insight in the execution behavior of the application is essent ...

Keywords: performance measurement and monitoring of java applications

19 Fragmentation considered harmful
Christopher A. Kent, Jeffrey C. Mogul
January 1995 **ACM SIGCOMM Computer Communication Review**, Volume 25 Issue 1

Full text available: pdf(1.25 MB) Additional Information: full citation, abstract, index terms

Internetworks can be built from many different kinds of networks, with varying limits on maximum packet size. Throughput is usually maximized when the largest possible packet is sent; unfortunately, some routes can carry only very small packets. The IP protocol allows a gateway to *fragment* a packet if it is too large to be transmitted. Fragmentation is at best a necessary evil; it can lead to poor performance or complete communication failure. There are a variety of ways to reduce the lik ...

20 A collaborative framework for distributed microscopy
B. Parvin, J. Taylor, G. Cong
November 1998 **Proceedings of the 1998 ACM/IEEE conference on Supercomputing (CDROM)**
Full text available: pdf(613.03 KB) Additional Information: full citation, abstract, references, citings

This paper outlines the motivation, requirements, and architecture of a collaborative framework for distributed virtual microscopy. In this context, the requirements are specified in terms of (1) functionality, (2) scalability, (3) interactivity, and (4) safety and security. Functionality refers to what and how an instrument does something. Scalability refers to the number of instruments, vendor-specific desktop workstations, analysis programs, and collaborators that can be accessed. Interactivi ...

Results 1 - 20 of 200 Result page: **1** 2 3 4 5 6 7 8 9 10 next